

False shares in verifiable secret sharing with finite field commitments

Hua Lu and Jack Peterson*
Forecast Foundation, San Francisco, CA 94107 USA

Verifiable secret sharing (VSS) is designed to allow parties to collaborate to keep secrets. We describe here a method of fabricating false secret shares that appear to other parties to be legitimate, which can prevent assembly of the decryption key. This vulnerability affects VSS schemes using verification commitments bounded to a finite field.

Verifiable secret sharing (VSS) schemes [1–3] rely on the assumption that parties can not reliably fabricate false secret shares which pass the verification process. Here, we show that, for certain VSS implementations that use verification commitments bounded to a finite field, this assumption is incorrect.

Let there be n shares of a secret and a threshold of t secret shares required to get the secret. Choose a finite field \mathbb{Z}_p and a generator $g \in \mathbb{Z}_p$. Each party associates with a unique non-zero identity $i \in \mathbb{Z}_p$ and creates a secret random polynomial,

$$P_i(z) = a_{i,0} + a_{i,1}z + \cdots + a_{i,t-1}z^{t-1}, \quad (1)$$

with coefficients $a_{i,j} \in \mathbb{Z}_p$, and decryption key given by $\sum_i a_{i,0}$. Each party also computes *verification commitments* $(c_{i,j})$,

$$c_{i,j} \equiv g^{a_{i,j}} \pmod{p}, \quad (2)$$

which are made available to all parties.

Next, each party computes and sends the k^{th} secret share, $P(k)$, to party k . Party k checks the incoming share against the sending party's verification commitments:

$$g^{P_i(k)} = g^{\sum_{j=0}^{t-1} a_{i,j} k^j} = \prod_{j=0}^{t-1} g^{a_{i,j} k^j} = \prod_{j=0}^{t-1} c_{i,j}^{k^j}. \quad (3)$$

Party k assumes the share received from party i is legitimate if it matches the verification commitments (available to all parties).

We now describe the vulnerability. First, note the following:

1. The polynomial interpolation for recovering decryption key parts must occur in the field \mathbb{Z}_p .
2. The multiplicative order of g divides $p - 1$.

Suppose one of the parties is an adversary. The adversary creates polynomial P_i and posts verification commitments $c_{i,j}$. The adversary sends false secret shares $Q_k \neq P_i(k)$ such that

$$Q_k = P_i(k) \pmod{p-1}. \quad (4)$$

The false shares check out, $g^{Q_i} = g^{P_i(k)}$; however, attempts to recover $P_i(0)$ using Q_k will most likely fail. The adversary can now prevent the decryption key from being assembled simply by not sharing $P_i(0)$.

In principle, this problem is avoidable by using verification commitments which are not bounded to the finite field. For example, we are free to choose commitments $c_{i,j} \equiv g^{a_{i,j}}$. However, note that $a_{i,j}$ is then on the order of p . Since this is at least 1024 bits, there is no practical way to store these verification commitments.

Financial support for this work was provided by J. Costello.

-
- [1] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
[2] G.R. Blakley. Safeguarding cryptographic keys. *National Computer Conference*, pages 313–317, 1979.
[3] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. *IEEE Symposium on Foundations of Computer Science*, pages 427–437, 1987.

*Electronic address: jack@augur.net